(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 599 449 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **93306833.0**

(51) Int. Cl.[5]: **G06F 15/16**

(22) Date of filing: **27.08.93**

(30) Priority: **27.11.92 JP 318212/92**

(43) Date of publication of application:
01.06.94 Bulletin 94/22

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **FUJITSU LIMITED**
1015, Kamikodanaka
Nakahara-ku
Kawasaki-shi Kanagawa 211(JP)

(72) Inventor: **Ueno, Haruhiko, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)
Inventor: **Nagasawa, Shigeru, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)
Inventor: **Ikeda, Masayuki, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)
Inventor: **Shinjo, Naoki, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)
Inventor: **Ishizaka, Ken-ichi, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)
Inventor: **Utsumi, Teruo, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)
Inventor: **Dewa, Masami, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)
Inventor: **Kobayakawa, Kazushige, c/o Fujitsu Limited**
1015 Kamikodanaka,
Nakahara-ku
Kawasaki-shi, Kanagawa 211(JP)

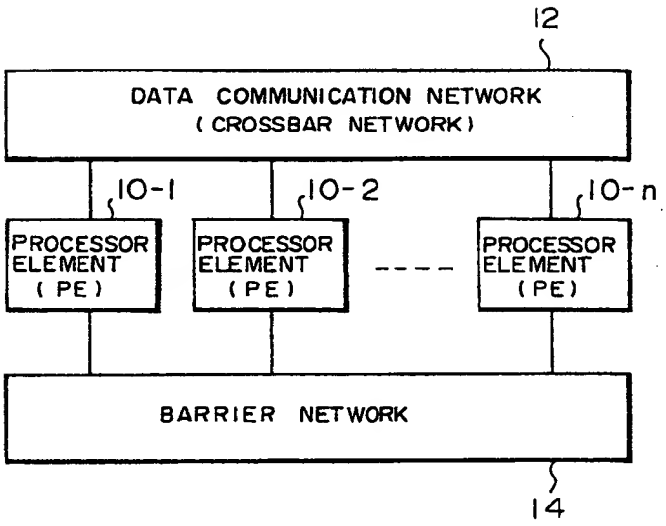
(74) Representative: **Billington, Lawrence Emlyn et al**
HASELTINE LAKE & CO
Hazlitt House
28 Southampton Buildings
Chancery Lane
London WC2A 1AT (GB)

(54) Data communication system and method.

(57) Any two multiple processor elements are coupled via a data communication network having a fixed communication buffer length and including multiple communication buffers. A packet of processed data having a header and body is created and transferred. Then a transmitting unit transmits dummy data, whose body is longer than the fixed communication buffer length, to the same receiving station. The transmitting unit then guarantees the receiving station the arrival of preceding processed data and the header. Control data representing cache invalidation waiting is embedded in the header of dummy data. When the transmitting end terminates transfer of dummy data, the sending station guarantees the termination of cache invalidation, which is performed by the receiving station to attain consistency of the contents of preceding storage data between a main storage and a cache memory.

EP 0 599 449 A2

FIG. 1


12
DATA COMMUNICATION NETWORK
(CROSSBAR NETWORK)
10-1
10-2
10-n
PROCESSOR ELEMENT (PE)
PROCESSOR ELEMENT (PE)
PROCESSOR ELEMENT (PE)
BARRIER NETWORK
14

The present invention relates to a data communication system and method for communicating processed data between processor elements coupled with each other via a data communication network. More particularly, this invention is concerned with a data communication system and method that guarantees the communication of processed data between processor elements constituting a parallel-processing machine.

In the field of computers, the adoption of the art of parallel processing aims at remarkable improvement in throughput. In a computer operating as a parallel-processing machine, a plurality of processing elements are coupled with one another via a data communication network such as a crossbar network including communication buffers. Unprocessed and processed data are transferred between processor elements. At this time, synchronous control is carried out to confirm the termination of data transfer between the processor elements and to proceed to the next parallel processing. The synchronous control makes it necessary to confirm that transfer data has reached a receiving station. It is therefor a must for improvement of the performance of a parallel-processing machine to guarantee data communication with a simple hardware configuration at high speed.

In a data communication system for transferring data via a data communication network such as a crossbar network including multiple buffers, the communication buffer length or a maximum communication buffer length for transmitting data from a transmitter to a receiver has been determined on a fixed basis but the transfer time is not always fixed. The reason why the transfer time is not determined on a fixed basis is that even when a receiver is coupled with a transmitter via a data communication network, if the receiver is busy, it cannot receive data. When a data communication network is a network made up of crossbar switches, if multiple transmitters send data to the same receiver, data from specific transmitters approach the receiver in the specified order of priorities. At some time instant, data transfer proceeds only for the data from one transmitter according to a defined order of priorities, and the other transmitters are forced to wait for the completion of the data transfer. As mentioned above, when a communication buffer length is fixed but a transfer time is not fixed, it must be guaranteed that data from a transmitter has reached a receiver. In a conventional method for providing this guarantee, the receiver uses a borrowed line to return a reception end signal to the transmitter or the receiver uses a data communication network to return a reception end signal to the transmitter.

However, in the method that a receiver uses a borrowed control line to return a reception end signal to a transmitter, more hardware is required to return a reception end signal. In the method that a receiver uses a data communication network to return a reception end signal, returnable data that represents termination of reception must be created at a receiving end. When hardware is used to create the returnable data, circuits become complex and large in scale. When software is used to create the returnable data, a prolonged period of processing time becomes necessary. In either of the methods, a time lag occurs after a receiver completes data reception until a transmitter is informed of termination of reception. This restricts the performance of a parallel-processing machine having numerous processor elements.

The present invention provides a data communication system and method such that when the communication data length in a data communication network is fixed but the transfer time is not fixed, it can be guaranteed without a time lag but with only a small number of hardware devices that transfer data from a transmitter has reached a receiver.

According to a first aspect of the present invention there is provided a data communication system, comprising:

data communication network means having a predetermined communication buffer length;

transmitting means for transmitting processed data;

receiving means for receiving data from said transmitting means; and

at least one communication guarantee means operable to transmit dummy data being longer than the communication buffer length, to the same receiving means as the one to which preceding processed data has been transmitted, immediately after the transmitting means has transmitted the preceding processed data;

whereby the arrival of the preceding processed data, sent from the transmitting means, at the receiving means is guaranteed.

According to a second aspect of the invention there is provided a data communication method in which a plurality of processing units, each of which has a main storage means and an instruction processing means and executes parallel processing, communicate with one another via a data communication network means that includes a plurality of communication buffers and has a definite communication buffer length, comprising:

a processed data transmitting step at which a first processing unit having a transmission means transmits processed data to any other processing unit that is designated as a receiving station; and

a communication guaranteeing step of transmitting dummy data, which is longer than the communication buffer length in said data communica-

tion network means, to the designated receiving station immediately after the processed data is transmitted at said processed data transmitting step;

whereby the arrival of processed data sent from said first processing unit at said receiving station is guaranteed.

According to a third aspect of the present invention there is provided a data system, comprising:

a plurality of processing units each of which has a main storage means and an instruction processing means and which plurality is arranged to execute parallel processing;

a data communication network means arranged to couple any two of said plurality of processing units via a plurality of communication buffers and having a predetermined communication buffer length;

each processing means having a transmission means arranged to transmit processed data to any other processing unit serving as a receiving station;

each processing means further having receiving means arranged to receive data from any other processing unit; and

communication guarantee means arranged to transmit dummy data having a data length greater than the predetermined communication buffer length to the receiving station immediately after a transmission means has transmitted the processed data;

whereby the arrival of the processed data at a processing unit serving as a receiving station is guaranteed.

Thus, the present invention is concerned with a parallel-processing machine comprising multiple processing elements (processing units) for executing processing on a parallel basis, a data communication network through which two of the multiple processor elements are coupled with each other with multiple communication buffers between them and which has a predetermined fixed communication buffer length, and a barrier network for synchronising parallel processing performed by multiple processor elements. In the parallel-processing machine according to the present invention, each of the multiple processing elements includes a transmitting unit that designates other processing elements as a receiving station and transmits processed data, which is stored in a main storage, to the receiving station, a receiving unit that receives transfer data from a transmitting unit in a specific processor element, and a communication guarantee unit for transmitting dummy data, which is longer than the fixed communication buffer length, to the same receiving station immediately after a transmitting unit transmits processed data. A processor element serving as a sending station thus guarantees that processed data has reached a processor element serving as a receiving station.

Any processor element may create and transfer a packet made up of a header and a body. The communication guarantee unit transmits dummy data formed as a packet composed of a header and a body with the data length in the packet made larger than the communication buffer length in the data communication network. A processor element serving as a sending station thus guarantees that preceding processed data has reached a receiving unit in a processor element serving as a receiving station. Since dummy data is transmitted with the data length of the body thereof made larger than the communication buffer length in the data communication network, a processor element serving as a sending station can guarantee how preceding processed data and the header of dummy data act on a receiving unit in a processor element serving as a receiving station. For this purpose, the communication guarantee unit embeds control data, which designates an operation to be performed on the preceding processed data by a processor element serving as a receiving station in the header of dummy data and then transmits the dummy data.

More particularly, the communication guarantee unit sets the data length LD_0 of a body in dummy data to a data length $LD_1 + LD_2$; that is, a sum of the communication buffer length LD_1 in the data communication network and a transfer data length LD_2 corresponding to the time required for the stoppage of data reception, which prevents transmission of a dummy packet from terminating until a processing unit serving as a receiving station interprets control data in the header of dummy data, and executes and terminates a designated operation before receiving by the body of dummy data. The transfer data length LD_2, for example, corresponds to the processing time required until a processor element serving as a receiving station stops data reception to start a designated operation after interpreting control data in a header. Thereby, a processor element serving as a sending station can guarantee the arrival of processed data at the processor element serving as a receiving station, the arrival of the header of dummy data, and the stoppage of data reception for performing operations designated in control data in the header.

More particularly, each of processor elements, which act as a parallel-processing machine, includes a cache memory that is installed in an instruction processing unit and stores the same processing data as those written in a main storage; a storage consistency control unit that when processed data is written in the main storage, compares the processed data with an address already registered in the cache memory, that when the data agrees with the address, queues the address

as an invalidation address for use in invalidating the old data in the cache memory, that dequeues such addresses sequentially, and then executes cache invalidation to maintain the consistency between the contents of the main storage and cache memory; a cache invalidation waiting unit that stops the data reception for the main storage while the storage consistency control unit is executing cache invalidation; and a control data receiving unit that when Interpreting the header of received data as a cache invalidation wait instruction, operates the cache invalidation waiting unit to stop data reception and allows the storage consistency control unit to execute cache invalidation.

The communication guarantee unit embeds control data, which represents cache invalidation waiting to a processor element serving as a receiving station, in a header of dummy data, and sets the data length LD_0 of a body of the dummy data to a data length $LD_1 + LD_2$; that is, a sum of the communication buffer length LD_1 in a data communication network and a transfer data length LD_2 corresponding to the processing time required until a processor element serving a receiving station interprets the control data in the header and stops data reception for buffer invalidation waiting. Consequently, each transmitting unit can guarantee:

I. the arrival of processed data at a processor element serving as a receiving station;

II. the arrival of the header of dummy data; and

III. the writing into a main storage and the termination of cache invalidation, which are operations designated in the header to be performed on preceding transfer data.

Each processor element includes a transmission end reporting unit that detects the termination of transmission of dummy data and issues a transmission end report to a barrier network. When receiving transmission end reports from all processor elements that have executed parallel processing, the barrier network extends control including synchronous control in which control is passed to the next step of parallel processing.

Each of processor elements which act as a parallel-processing machine includes a main storage, a main control unit, a scalar unit including a central processing unit for executing scalar computation and a cache memory, a vector unit for executing vector computation, a data transmission unit (data transmission processor) for transferring data to or from a data communication network, a synchronous processing unit (barrier processing unit) for extending synchronous control via a barrier network, and an input/output unit for inputting or outputting data from or to external equipment. If necessary, a processor element may not include the vector unit and/or input/output unit. The data communication network for coupling multiple processing elements on a one-to-one basis is a crossbar network including buffer memories.

According to the foregoing data communication system of the present invention, when a transmitting end detects the termination of transmission of dummy data which is longer than the communication data length in the data communication network, it can be guaranteed that all the data transmitted previously have arrived at a receiving station. Particularly, a packet composed of a header containing various kinds of control information for the receiving station and a body containing processed data itself is created and then transferred as transfer data for one communication session. The data length in the body of dummy data is larger than the communication buffer length, whereby when a transmitting end terminates transmission of dummy data, it can be guaranteed that the preceding storage data and the header of the dummy data have arrived at a receiving station. When control data representing a specified operation to be performed by a receiving unit in a receiving station is embedded in the header of dummy data, if the data length in the body of dummy data formed as a packet is a sum of:

Communication buffer length + Transfer data length corresponding to the time required until the control data in the header of dummy data is interpreted and a designated operation is started

With the termination of transmission of dummy data at a transmitting end, a receiving end can therefore guarantee the termination of all operations needed for preceding transfer data and pass control to the next processing.

In the aforesaid data communication system according to the present invention, store access to a main storage for data transferred to the receiving station and cache invalidation for a cache memory are carried out to maintain the consistency between the contents of the main storage and cache memory, so that a sending station can guarantee the processing of a receiving station. To be more specific, a processor element that has received transfer data performs storage access to write the transfer data in a main storage, and executes cache invalidation to invalidate old data existent in the cache memory. The cache invalidation is referred to as buffer invalidation (BI). The buffer invalidation is executed by a storage consistency control circuit in a main control unit (MCU) incorporated in a processor element. The storage consistency control circuit includes a memory, which is referred to as a second tag field TAG2, for storing registered addresses of the cache memory. A storage address is compared with a cache address contained in the second tag field TAG2. When the addresses agree with each other, a cache invalidation address (BI address) is generated. In contrast,

a cache memory installed in a central processing unit comprises a first tag field TAG1, in which registered cache addresses are stored, and a data division. A registration address in the first tag field TAG1, which corresponds to the cache invalidation address generated by the main control unit, is invalidated. After the main storage is rewritten, access to old data in the cache memory is inhibited. As a result, the contents of the main storage and cache memory become consistent with each other.

Invalidation of the first tag field TAG1 according to cache invalidation addresses generated by the main control unit is not executed immediately but temporarily placed in a stack memory which is referred to as a BI queue (BI addressing queue). Before the BI queue becomes full, data reception as part of store access is stopped, and cache invalidation addresses are fetched sequentially from the BI queue. Invalidation of the first tag TAG1 is then executed. This kind of control for matching the contents of the main storage and cache memory is also performed when storage data is received from other processor element.

The data communication system according to the present invention can guarantee that when a transmitting end terminates transmission of dummy data, a processor element serving as a receiving end terminates the cache invalidation for storage consistency control. For this guarantee, control data representing cache invalidation waiting at the transmitting end is embedded in the head of dummy data, and then the dummy data is transferred. The control data in the header, which represents cache invalidation waiting, is interpreted by a control data receiving unit. A cache invalidation waiting unit is then activated. The cache invalidation waiting unit stops reception of storage data in order to execute cache invalidation, waits for buffer invalidation to terminate, and then restarts data reception.

With the operation of cache invalidation waiting performed by a receiving station, the time required until data reception is started is guaranteed by stopping data reception until a receiving unit terminates cache invalidation. When transmission of dummy data terminates, the receiving station executes buffer invalidation waiting to temporarily stop data transfer. All cache invalidation addresses indicating previous storage data are then read from a BI queue sequentially. Cache invalidation is then executed to invalidate the addresses in the first tag field TAG1. The BI queue is thus emptied. Cache invalidation terminates (the consistency between the contents of the main storage and cache memory is attained). Cache invalidation waiting is then released to restart data reception.

Data may be distributed to numerous processor elements and subjected to parallel processing. In this case, when a transmitting end confirms that transfer of dummy data has terminated immediately after transfer of processed data, it can be guaranteed that a receiving station has terminated the store access for a main storage and cache invalidation. Using the termination of the transmission of dummy data by all processor elements as synchronous information, control can be passed to the parallel processing at the next step at a high speed.

For a better understanding of the invention and to show how the same may be carried into effect reference will now be made, purely by way of example, to the accompanying drawings in which:-

Fig. 1 is a block diagram of a parallel-processing machine in which a data communication system of the present invention is implemented;

Fig. 2 is a schematic explanatory diagram of the data communication network in Fig. 1;

Fig. 3 is a block diagram showing an embodiment of a processor element in Fig. 1;

Fig. 4 is a block diagram showing another embodiment of a processor element in Fig. 1;

Fig. 5 is an explanatory diagram showing a global memory in the parallel-processing element in Fig. 1 and parallel processing;

Fig. 6 is a block diagram showing an embodiment of a main control unit installed in a processor element;

Fig. 7 is a block diagram showing the details of the scalar unit in Fig. 6;

Fig. 8 is an explanatory diagram showing a state of communication between elements according to the present invention;

Fig. 9 is an explanatory diagram showing a state in which dummy data used for the communication in Fig. 8 is stored in a main storage;

Fig. 10 is a timing chart showing data communication between the elements in Fig. 8; and

Fig. 11 is a timing chart showing fundamental data communication using dummy data according to the present invention.

Fig. 1 schematically shows a parallel-processing machine in which a data communication system of the present invention is implemented. In this parallel-processing machine, n units of processor elements 10-1, 10-2, etc., and 10-n are connected with one another via a data communication network 12 such as a crossbar network consisting of multiple communication buffers, so that they can be coupled with one another on a one-to-one basis. The parallel-processing machine further includes a barrier network 14 for synchronizing parallel processing performed by the processor elements 10-1 to 10-n. Up to n = 222 units of processor elements can be included as the processor elements 10-1 to 10-n. For example, when the throughput of one processor element is 1.6 giga FLOPS, the processor elements 10-1 to 10-n provide a peak through-

put of 355.2 giga FLOPS as a whole.

In Fig. 2, the schematic configuration of the data communication network 12 in Fig. 1 is shown on the assumption that five units of processor elements 10-1 to 10-5 are installed. In Fig. 2, data communication is carried out from four processor elements 10-1 to 10-4 to one processor element 10-5. The data communication network 12 has a hierarchical structure in which the processor element 10-5 serving as a shaped receiving station ranks higher order than the four processor elements 10-1 to 10-4 serving as sending stations. In the hierarchical structure using transfer buffers 16-1 to 16-7 of this embodiment shown in Fig. 2, a communication path has established between the processor elements 10-1 and 10-5. Data transfer requested by the other processor elements 10-2 to 10-4 is placed in the wait state intermediately. Specifically, the processor element 10-1 has accessed the buffers 16-1, 16-5, and 16-7 to establish a communication path leading to the processor element 10-5, and is carrying out data transfer. In contrast, the processor element 10-2 succeeds in accessing the buffer 16-2 but fails in accessing the buffer 16-5 because of priority control. After transferring data to the buffer 16-2, the processor element 10-2 is placed in the wait state. The processor element 10-3 succeeds in accessing the transfer buffer 16-3 but fails in accessing the transfer buffer 16-6 because of the priority control by the processor element 10-4. The processor element 10-3 has stopped data transfer after transferring data up to the transfer buffer 16-3. The processor element 10-4 succeeds in accessing the transfer buffer 16-6 but fails in accessing the transfer buffer 16-7 in the last stage because the transfer buffer 16-7 has already been accessed by the processor element 10-1. The processor element 10-4 is thus placed in the wait state.

In data communication between processor elements via the data communication network, which consists of transfer buffers, shown in Fig. 2, the communication data length in the data communication network is determined on a fixed basis depending on the number of transfer buffers employed. However, the transfer time is not determined on a fixed basis because of the latency resulting from priority control of buffers.

Fig. 3 shows an internal configuration of each of the processor elements 10-1 to 10-n shown in Fig. 1. A processor element 10 comprises a main control unit (MCU) 18, a main storage (MSU) 20, a scalar unit 22, a vector unit 24, a data transmission processor 26 for transferring data via a data communication network 12, a barrier processing unit 28 for extending synchronous control to a barrier network 14, and an input/output subsystem 30 for extending input/output control to external storages. The scalar unit 22 includes a CPU 32 and a cache memory 34. The cache memory 34 consists of a data division 38 in which the same data as those existent in the main storage 20 are stored, and a first tag field TAG1 36 in which storage addresses of data stored in the data division 38 are contained. The main control unit 18 includes a second tag field TAG2 40 and a buffer invalidation queue (BI queue) in which buffer invalidation addresses are queued. The second tag field 40 in the main control unit 18 contains addresses including the addresses existent in the first tag field 36 in the cache memory 34 installed in the scalar unit 22. For example, the first tag field 36 contains addresses for, for example, four blocks on the assumption that one block consists of sixteen bytes. In contrast, the second tag field 40 contains block addresses as cache addresses on the assumption that one block consists of 256 bytes or a four-fold value of the number of bytes in one block in the first tag field 36.

When a processing unit other than the CPU 32 performs storage access on the main storage 20, the main control unit 18 references the second tag field 40 for a storage address. If a consistent address is present, the address is put as a buffer invalidation address (BI address) in the buffer invalidation queue 42. Invalidation addresses enqueued in the buffer invalidation queue 42 are sent to the CPU 32 according to the specified timing that the queue becomes full. If consistent addresses are present in the first tag field 36 in the cache memory 34, the consistent addresses are invalidated. When buffer invalidation is executed for the cache memory 34 by dequeuing buffer invalidation addresses from the buffer invalidation queue 42, the reception of storage data for the main control unit 18 is placed in the stopped state.

If the main storage 20 is a shared memory that is also used by a scalar unit installed in other main control unit, serialization is performed concurrently with buffer invalidation for the cache memory 34. During the serialization, the buffer invalidation addresses fetched from the buffer invalidation queue 42 are transferred to a buffer invalidation queue installed in other main control unit. Buffer invalidation is performed on a cache memory associated with a CPU that is connected to the above other main control unit. The contents of the main storage 20 and cache memory 34 thus become consistent with each other.

Fig. 4 shows another embodiment of a processor element shown in Fig. 1. This embodiment does not include the vector unit 24 and input/output subsystem 30 shown in Fig. 3. The processor element 10 having the configuration shown in Fig. 4 may be employed as one of processor elements constituting a parallel-processing machine. Even in

the processor element 10 in Fig. 4, the cache memory 34 in the scalar unit 22 consists of the first tag field (TAG1) 36 and data division 38. The main control unit 18 includes the second tag field (TAG2) 40 and buffer invalidation queue (BI queue) 42.

Fig. 5 shows a configuration of a memory consisting of main storages 20-1 to 20-n installed in processor elements 10-1 to 10-n that constitute a parallel-processing machine in Fig. 1. The main storages 20-1 to 20-n installed in the processor elements 10-1 to 10-n are used as local main storages in the respective processor elements. Portions of the main storages 20-1 to 20-n, which are hatched in Fig. 5, are assigned as shared memory areas 44-1 to 44-n. The shared memory areas 44-1 to 44-n are managed using logical addresses that are defined in units of multiple physical addresses of shared memory areas, thus forming a global memory 46 to be shared by all the processor elements 10-1 to 10-n. The global memory 46 consists of the multiple processor elements 10-1 to 10-n. Thanks to this configuration, after data read from the global memory 46 are distributed to the processor elements 10-1 to 10-n and parallel processing is executed, when the results of the processing are stored at a specific physical address in the global memory 46. During parallel processing using the global memory 46, data transfer is performed between processors via the data communication network.

Assuming that five processor elements execute a DO loop of DO (0, 100) as parallel processing, the parallel processing of the DO loop nested to I = 20 levels is allocated to each of the five processor elements. Data needed for the DO loop at a level is read from the shared memory area in a specific processor element, and allocated to the processor element concerned via the data communication network 12. The five processor elements thus execute the DO loop by processing 20 nested DO loops on a parallel basis. When each processor element finishes with the allocated DO loops, used data are transferred to and written in the shared memory areas in the specific processor elements in which the data used to reside. The barrier network 14 checks termination of writing. Control is then passed to the parallel processing of a subsequent DO loop. As mentioned above, parallel processing is executed by distributing data from specific areas in the global memory 46 to the multiple processor elements 10-1 to 10-n. This kind of parallel processing required such data communication that processed data are transferred to and written in the areas in the global memory 46 in which the data used to reside, and, at the same time, the completion of the storage consistency control, which attains the consistency between the contents of the global memory 46 and each of cache memories by performing invalidation on the cache memories, is guaranteed.

Fig. 6 shows the details of a main control unit installed in a processor element of the present invention. The main control unit 18 has vector unit access pipe lines 60-1, 60-2, 60-3, and 60-4 leading to the vector unit 24. In this embodiment, the vector unit 24 executes four operations of pipe line processing for a mask pipe line, a multiplication pipe line, an addition/logical computation pipe line, and a division pipe line. The four vector unit access pipe lines 60-1 to 60-n are therefore associated with the four operations of pipe line processing. Storage consistency control units 64-1 to 64-4 are included in association with the vector unit access pipe lines 60-1 to 60-4. Each of the storage consistency control units 64-1 to 64-4, for example, the storage consistency control unit 64-1 includes a second tag field 70-1 and a buffer invalidation queue 72-1.

A scalar unit access pipe line 62 and a storage consistency control unit 66 are included to cope with store access which the scalar unit 22 and data transmission processor 26 are requested of by other processor elements coupled via the data communication network 12. The storage consistency control unit 66 has a second tag field 40 in association with the first tag field 36 in the cache memory 34 in the scalar unit 22. The storage consistency control unit 66 further includes a buffer invalidation queue 42 so that when a store access address or an address at which store access is requested agrees with an address in the second tag field 40, the address is queued as a buffer invalidation address.

Each of the vector unit access pipe lines 60-1 to 60-4, scalar unit access pipe line 62, and storage consistency control units 64-1 to 64-4, and 66 receives a store access request via a priority order circuit 50. The priority order circuit 50 is coupled with the four access paths extending from the vector unit 24 via an interface 48. The priority order circuit 50 inputs a store access request from the scalar unit 22 via an interface 52 which is selected by a selector 56, or from any other processor element through the data transmission processor 26 via an interface 52. Any of buffer invalidation addresses provided by the storage consistency control unit 66 in the scalar unit and by the storage consistency control units 64-1 to 64-4 in the vector unit are selected by a selector 68, and fed to the CPU 32 in the scalar unit 22. Based on the selected addresses, buffer invalidation is performed on the cache memory 30; that is, addresses in the first tag field 36 are invalidated.

Fig. 7 shows the details of the scalar unit access pipe line 62 and storage consistency control unit 66 shown in Fig. 6. The scalar unit access

pipe line 62 comprises multiple registers of a register 74 to a register 84 in the last stage, wherein the multiple registers are coupled with one another by multistage connection. From the register 74 in the first stage to the register 84 in the last stage, for example, access data is transmitted at a 15 clock cycle. During the transfer from the register 80 to the register 82 located in the middle of the pipe line, store access or load access is executed for the main storage 20. In the storage consistency control unit 66, the output of a register 86 is branched into registers 88 and 90. The second tag field 40 is installed in a stage succeeding the register 88. In the second tag field 40, address data are stored as block addresses for blocks each consisting of, for example, 256 bytes by store access. Addresses are fetched from the second tag field 40 into a register 92 in synchronization with the transfer of data for store access to a register 94. A comparator 96 determines whether or not the data agree with the addresses. If the addresses for store access agree with the addresses existent in the second tag field 40, the comparator 96 instructs the buffer invalidation queue 42 to carry out storage control. The storage addresses in the register 94 are queued as buffer invalidation addresses in the buffer invalidation queue 42.

The state of comparison for determining a buffer invalidation address to be queued in the buffer invalidation queue 42 is supervised by a cache invalidation control unit 98. When a specified number of addresses, the number of addresses which have not fill the buffer invalidation queue 42, have been stacked in the buffer invalidation queue, the cache invalidation control unit 98 outputs a buffer invalidation signal to the CPU 32. At the same time, the cache invalidation control unit 98 activates a cache invalidation waiting unit 100. The cache invalidation waiting unit 100 outputs an inhibiting signal to the interfaces 52 and 54 in Fig. 6, thus stopping the data reception from the scalar unit 22 or any other processor element for the main storage 18.

As mentioned above, the cache invalidation control unit 98 stops reception of storage, data, and then fetches buffer invalidation addresses sequentially from the buffer invalidation queue 42. The cache invalidation control unit 98 then transfers the addresses to the selector 68 and to the CPU 32 via a register 104. Consequently, buffer invalidation is performed on the cache memory 34; that is, addresses in the first tag field 36 are invalidated. Buffer invalidation addresses existent in the register 104 are interpreted by a decoder 106. The decoder 106 generates serialization signals to other main control units.

In this embodiment, only one main control unit is installed for the main storage 20. Serialization, which is required when multiple main control units access a single main storage, therefore need not be done. However, there is a possibility that a single main storage 20 may be shared by multiple main control units 18 installed in a single processor element. Hardware is therefore installed preliminarily to assist the decoder 106 in outputting serialization signals to other main control units.

Fig. 8 is an explanatory diagram showing the processor elements 10-1 and 10-n to reveal the facilities required for a data communication network of the present invention, wherein data is transferred from the processor element 10-1 to the processor element 10-n. The facilities of a communication guarantee control unit 108 and a transmission end reporting unit 110 are validated for a main control unit 18-1 in the processor element 10-1 serving as a transmitting end. On the other hand, the facilities of a control data receiving unit 102, a storage consistency control unit 66, and an access pipe line 62 are validated for a main control unit 18-n in the processor element 10-n serving as a receiving end. The data communication network 12 carries out data transfer within a communication buffer length defined by the transfer buffers 16-1 to 16-n.

Fig. 9 shows the main control units 20-1 and 20-n in the processor elements 10-1 and 10-n in the state of data communication. During data communication according to the present invention, the processor element 10-1 serving as a transmitting end creates a packet made up of a header and a body, and then transfers storage data to the processor element 10-n serving as a receiving station. According to the present invention, the transfer of storage data is succeeded by the transfer of a packet of dummy data for use in guaranteeing the operation of a receiving end with the processing done by a transmitting end alone. A packet of dummy data is, similarly to a packet of normal data, composed of a header and a body. The data length in the body is provided as a sum of the communication buffer length in the data communication network 12 and the transfer data length corresponding to the processing time required until the control data receiving unit 102 in the processor element 10-n serving as a receiving station interprets the header of the packet of dummy data and the buffer invalidation waiting unit stops data reception. The body of a packet of dummy data, which is used to guarantee the operations of the processor element 10-n serving as a receiving station that end with the termination of buffer invalidation for all storage data, is stored beforehand in each of dummy data storages areas 114-1 and 114-n in the main storages 20-1 and 20-n. Data embedded in the body of dummy data and stored in the dummy data storage areas 114-1 and 114-n is so-called garbage that is meaningless to any processing in

the respective processor elements. However, the data length of garbage is significant in guaranteeing a period of time required for completing the operations of a receiving station ending with buffer invalidation. When the communication buffer length in the data communication network 12 varies with a processor element serving as a receiving end, the data length of so-called garbage stored in each of the dummy data storage areas 114-1 and 114-n must be changed. Specifically, data having a different data length and including an address pointer for indicating the location of a receiving station is stored in the dummy data storage areas 114-1 and 114-n.

Fig. 10 is a timing chart showing the temporal transition of data communication from the processor element 10-1 to processor element 10-n shown in Figs. 8 and 9. At the time instant t_1, storage data 120 which has been processed by the processor element 10-1 serving as a transmitting end is available the form of a packet having a header 122 and and a body 124. At the same time, dummy data 130 is also available in the form of a packet having a header 132 and a body 134 and succeeding the storage data 120. The header 132 of the dummy data 130 contains the control data for buffer invalidation waiting to be sent to the storage consistency control unit in the main control unit 18-n in the processor element 10-n serving as a receiving end. The data length DL_0 in the body 134 of the dummy data 130 is provided as a sum of the data length DL_1 in the data communication network 12 and the transfer data length DL_2 corresponding to the time required until the processor element 10-n serving as a receiving end receives and interprets the header 132 of the dummy data 130 and the buffer invalidation waiting unit stops data reception; that is,

$$DL_0 = DL_1 + DL_2$$

At the time instant t_2 in Fig. 10, data transfer of the preceding storage data 120 starts, and the header 122 reaches the entry of the processor element 10-n serving as a receiving end. In this explanation, the transmission of the body 124 of the preceding storage data 120 to the data communication network 12 terminates at the same time. At the time instant t_2 in Fig. 10, when the transmission of the preceding storage data 120 to the data communication network 12 terminates, the dummy data 130 is transmitted immediately with the communication path of the data communication network 12 occupied.

At the time instant t_3, the leading end of the dummy data 130 transmitted immediately after the storage data 120 reaches the processor element 10-n serving as a receiving end. In this state, the header of the preceding storage data 120 has been interpreted, addresses for store access for a main storage and for buffer invalidation have been produced, and consistent addresses have been stacked in the buffer invalidation queue.

At the time instant t_4, transmission of the dummy data 130 to the data communication network 12 terminates. In the processor element 10-n serving as a receiving end, the control data for buffer invalidation waiting contained in the header 132 of the dummy data 130 is interpreted by, for example, the control data receiving unit 102 in the storage consistency control unit 66 in Fig. 7. A control signal is output to the cache invalidation waiting unit 100 according to the interpreted control data for buffer invalidation waiting. In response to the cache invalidation control signal sent from the control data receiving unit 102, the cache invalidation waiting unit 100 outputs an inhibiting signal to, for example, each of the interfaces 52 and 54 shown in Fig. 6, and stops the data reception from the scalar unit 22 and data transmission processor 26. In the state attained at the time instant t_4 in Fig. 10, data transfer of the dummy data 130 to the processor element 10-n stops. In synchronization with stop control of data reception, the cache invalidation waiting unit 100 shown in Fig. 7 instructs the cache invalidation control unit 98 to activate invalidation. The cache invalidation control unit 98 is not be bound to the conditions for activation, which depend on the number of addresses normally enqueued in the buffer invalidation queue 42, but outputs a buffer invalidation signal to the CPU 32 in order to request cache invalidation control. At the same time, the cache invalidation control unit 98 reads buffer invalidation addresses sequentially from the cache invalidation queue 42, supplies the read addresses to the CPU 32, and then executes the invalidation of the cache memory 32; that is, invalidates addresses in the first tag field 36. When completing the dequeuing of all the cache invalidation addresses from the buffer invalidation queue 42, the cache invalidation control unit 98 reports termination of processing to the cache invalidation waiting unit 100. In response to the report, the cache invalidation waiting unit 100 releases the inhibiting signals sent to the interfaces 52 and 54 shown in Fig. 6. Consequently, transfer of dummy data from the processor element 10-1 serving as a transmitting end via the data transmission processor 26, which has been in the stopped state, is restarted.

At the time instant t_4 in Fig. 10, termination of transmission of the dummy data 130 from the processor element 10-1 serving as a transmitting end to the data communication network 12 is detected by the transmission control unit 108 in the main control unit 18-1 in the processor element 10-

1 serving as a transmitting end in Fig. 8. The transmission end reporting unit 110 reports the termination of transmission of dummy data to the barrier processing unit 28-1. In response to the termination end report, the barrier processing unit 28-1 outputs a synchronization request signal to the barrier network 14 so that control will be passed to the parallel processing at the next step.

Data communication from the processor element 10-1 to processor element 10-n shown in Fig. 10 is also performed between other processor elements. When termination of transmission of dummy data is detected, each processor element outputs a synchronization request signal to the barrier network 14. With the input of synchronization request signals sent from all processor elements 10-1 to 10-n, the barrier network 14 recognizes that parallel processing, which has been distributed to the processor elements, has terminated and that the conditions for transition to the next parallel processing stored in the main storages forming the global memory have been established. The barrier network 14 then executes synchronous control in order to allocate the next parallel processing to the processor elements.

After data reception stops at the time instant t_4, when the data reception restarts with the completion of buffer invalidation, the received dummy data 130; that is, the body 134 of the dummy data 130 is stored as garbage in the dummy data storage area 114-n in the main storage 20-n in the processor element 10-n serving as a receiving end as shown in Fig. 9.

Fig. 11 is a timing chart showing the operations of communication to be done when a data communication system of the present invention is further simplified. In the aforesaid embodiments, a data communication system of the present invention applies to data transfer within a parallel-processing machine. The present invention is not limited to this data transfer. When data transfer is performed between two data processing units via transfer buffers with a data length fixed but the transfer time unfixed, reception of transfer data by a receiving end can be guaranteed with the processing done by a transmitting end alone.

In Fig. 11, storage data 120 is transmitted from a transmitting processor element to a receiving processor element via the data communication network 12. At the time instant t_1, the dummy data 130 whose data length equals to the data length DL_1 in the data communication network 12 is made available. First, at the time instant t_2, the storage data 120 is transmitted. When the transmission of the storage data 120 terminates, the dummy data 130 is transmitted immediately. In the state attained at the time instant t_3 at which the transfer of the dummy data 130 terminates, since the data length DL_0 in the dummy data 130 equals to the data length DL_1 in the data communication network 12, when a transmitting end detects the termination of the transmission of the dummy data 130, it is guaranteed that the preceding storage data 120 have reached the receiving processor element without fail. In the data communication system according to the present invention, all that is required is to transmit the dummy data 130, in which the data length DL_0 is longer than the data length DL_1 in the data communication network 12, immediately after the preceding transfer data.

As described so far, according to the present invention, the arrival of transferred data at a receiving end and the carrying out of specified operations on received data can be guaranteed with the detection of termination of transmission of dummy data by a transmitting end. From the beginning of data transfer, only a limited period of time is needed to learn the completion of data transfer to a receiving end and activate the processing required with the termination of data transfer. What is required for providing the above guarantee is that a transmitting end transmits dummy data. Only a limited number of hardware devices are therefore required to be installed additionally. Furthermore, control data that represents the operations to be performed by a receiving end can be appended to the prefix of dummy data. This results in a novel function that not only completion of data communication but also subsequent operations concerning received data such as buffer invalidation control for attaining the consistency between the contents of a main storage and a buffer memory can be guaranteed with the termination of transmission of dummy data by a transmitting end.

The control data embedded in the prefix or header of dummy data is not limited to buffer invalidation control data described in the aforesaid embodiments but may be any control data that is helpful in guaranteeing the operations required to be performed by a processing element serving as a receiving end with the reception of preceding storage data.

In the aforesaid embodiments, data transfer is performed in order to access a global area. The present invention further applies to all the other kinds of data transfer between processor elements. Moreover, the present invention can be modified in various manners and will therefore not be restricted to the numerical values indicated in the embodiments.

Claims

1. A data communication system, comprising:
   data communication network means having a predetermined communication buffer length;

transmitting means for transmitting processed data;

receiving means for receiving data from said transmitting means; and

at least one communication guarantee means operable to transmit dummy data being longer than the communication buffer length, to the same receiving means as the one to which preceding processed data has been transmitted, immediately after the transmitting means has transmitted the preceding processed data;

whereby the arrival of the preceding processed data, sent from the transmitting means, at the receiving means is guaranteed.

2. A data communication system according to claim 1 wherein said transmitting means includes a transmission end reporting means that detects the termination of transmission of dummy data and reports it to other processing units.

3. A data communication method in which a plurality of processing units, each of which has a main storage means and an instruction processing means and executes parallel processing, communicate with one another via a data communication network means that includes a plurality of communication buffers and has a fixed communication buffer length, comprising:

a processed data transmitting step at which a first processing unit having a transmission means transmits processed data to any other processing unit that is designated as a receiving station; and

a communication guaranteeing step of transmitting dummy data, which is longer than the fixed communication buffer length, to the designated receiving station immediately after the processed data is transmitted at said processed data transmitting step;

whereby the arrival of processed data sent from said first processing unit at said receiving station is guaranteed.

4. A data communication method according to claim 3, wherein dummy data to be transmitted at said communication guaranteeing step is a packet composed of a header and a body, and transmitted with the data length in said packet made longer than the fixed communication buffer length.

5. A data communication method according to claim 3, wherein dummy data to be transmitted at said communication guaranteeing step is a packet composed of a header and a body, and transmitted with the data length of said packet made longer than the fixed communication buffer length.

6. A data communication method according to claim 3, wherein at said communication guaranteeing step, said dummy data is transmitted with control data embedded in the header thereof which designates operations to be performed on preceding processed data by said receiving station.

7. A data communication method according to claim 6, wherein at said communication guaranteeing step, the data length (DL_0) of said body of dummy data is set to a data length being a sum of the communication buffer length (DL_1) in said data communication network means and the transfer data length (DL_2) corresponding to the time required for the stoppage of data reception that prevents transmission of dummy data from terminating until said receiving station interprets control data in said header of dummy data, and executes and terminates designated operations before receiving said body of dummy data; and the arrival of processed data at said receiving station, the arrival of the header of dummy data, and the stoppage of data reception designated in control data in said header are guaranteed by the first processing unit.

8. A data communication method according to any one of claims 3 to 7,

wherein each of said processing units comprises:

a cache memory that is installed in an instruction processing means and stores the same processed data as those written in a main storage;

a storage consistency control means operable, when processed data is written in said main storage, to compare the processed data with a registered address of said cache memory, when the processed data agrees with the registered address, to queue the address as an invalidation address for use in invalidating old data in said cache memory, and to dequeue such addresses sequentially and then to execute cache invalidation to maintain the consistency between the contents of said main storage means and said cache memory;

a cache invalidation waiting means operable, while said storage consistency control means is executing cache invalidation, to stop the data reception for said main storage means; and

a control data receiving means arranged,

when interpreting the head of received data as a cache invalidation wait instruction, to operate said cache invalidation means to stop data reception and to allow said storage consistency control means to execute cache invalidation;

at said communication guaranteeing step, said dummy data is transmitted with control data, representing cache invalidation waiting embedded in the header thereof to said receiving station; the data length (LD_0) of said body of dummy data is provided as a sum of the communication buffer length (LD_1) in said data communication network means and the transfer data length (DL_2) corresponding to the processing time required until said processing unit serving as a receiving station interprets said control data in said header of dummy data and stops data reception for buffer invalidation waiting; and

whereby the arrival of processed data at said receiving station, the arrival of the header of dummy data, and the writing into said main storage means and the termination of cache invalidation, which are designated in said header as operations to be performed on preceding transfer data, are guaranteed at the restart of data reception.

9. A data communication method according to any one of claims 3 to 8, further including a transmission end reporting step of detecting termination of the transmission of dummy data executed at said communication guaranteeing step and issuing a transmission end report to a barrier network means; said barrier network means passing control to the parallel processing at the next step when receiving transmission end reports from all of said processing units that have executed parallel processing.

10. A data communication method according to any of claims 3 to 9, wherein said processing unit has a dummy data storage area, which stores meaningless data having a specified data length for use as the data length in the body of dummy data, in the main storage means thereof; and at said communication guaranteeing step, data transfer is performed using a storage address of said dummy data storage area.

11. A data communication method according to any one of claims 3 to 10,
wherein portions of the main storage means installed in said plurality of processing units are assigned as a global memory; and all of said processing units access the global memory consisting of said main storage means to execute parallel processing.

12. A data communication system, comprising:

a plurality of processing units each of which has a main storage means and an instruction processing means and which plurality is arranged to execute parallel processing;

a data communication network means arranged to couple any two of said plurality of processing units via a plurality of communication buffers and having a predetermined communication buffer length;

each processing means having a transmission means arranged to transmit processed data to any other processing unit serving as a receiving station;

each processing means further having receiving means arranged to receive data from any other processing unit; and

communication guarantee means arranged to transmit dummy data having a data length greater than the predetermined communication buffer length to the receiving station immediately after a transmission means has transmitted the processed data;

whereby the arrival of the processed data at a processing unit serving as a receiving station is guaranteed.

13. A data communication system according to claim 1 or claim 12 wherein dummy data to be transmitted by said communication guarantee means is a packet composed of a header and a body, and transmitted with the data length in said packet made longer than the predetermined communication buffer length; and whereby the arrival of at least one of preceding processed data and said header of dummy data at said processing unit serving as a receiving station is guaranteed by said processing unit serving as a sending station.

14. A data communication system according to any one of claims 1, 2, 12 or 13, whereby said communication guarantee means transmits said dummy data with control data embedded in the header thereof, which designates operations to be performed on preceding processed data by said receiving station.

15. A data communication system according to any one of claims 1, 2, 12, 13, or 14, wherein said communication guarantee means is arranged to set the data length (LD_0) of the body of said dummy data to a data length being a sum of the predetermined communication buffer length (LD_1) and the transfer data length

(LD_2) corresponding to the time required for the stoppage of data reception that prevents transmission of dummy data from terminating until said receiving station has interpreted control data in said header of dummy data, and executed and terminated specified operations before receiving the body of dummy data; and whereby the arrival of processed data at said receiving station, the arrival of the header of dummy data, and the stoppage of data reception designated in the control data in said header are guaranteed by said processing unit serving as a sending station.

16. A data communication system according to any one of claims 1, 2 or 12 to 15, wherein each of said processing units comprises:

a cache memory that is installed in an instruction processing means and stores the same processed data as those written in a main storage;

a storage consistency control means that when processed data is written in said main storage means, compares the processed data with a registered address of said cache memory, that when the processed data agrees with the registered address, queues the address as an invalidation address for use in invalidating old data in said cache memory, and that dequeues such addresses sequentially and then executes cache invalidation to maintain the consistency between the contents of said main storage means and said cache memory;

a cache invalidation waiting means that while said storage consistency control means is executing cache invalidation, stops the data reception for said main storage means; and

a control data receiving means that when interpreting the header of received data as control data representing cache invalidation waiting, operates said cache invalidation means to stop data reception and allows said storage consistency control means to execute cache invalidation;

said communication guarantee means embeds control data, which represents cache invalidation waiting to a processing unit serving as a receiving station, in the header of said dummy data, and sets the data length (LD_0) of said body of dummy data to a data length provided as a sum of the communication buffer length (LD_1) in said communication network means and the transfer data length (LD_2) corresponding to the processing time required until said processing unit serving as a receiving station interprets control data in said header of dummy data and stops data reception for buffer invalidation waiting; and

the arrival of processed data at said receiving station, the arrival of the header of dummy data, and the writing into said main storage means and the termination of cache invalidation, which are designated in said header as operations to be performed on preceding transfer data, are guaranteed at the restart of data reception.

17. A data communication system according to any one of claims 1 or 12 to 16 wherein at leasIt one processing unit includes a transmission end reporting means that is arranged to detect termination of transmission of dummy data and then to issue a transmission end report to a barrier network means; and said barrier network means is arranged to pass control to the parallel processing at the next step when receiving transmission end reports from all of said processing units which have executed parallel processing.

18. A data communication system according to any one of claims 1, 2 or 12 to 17 wherein at least one processing unit has a dummy data storage area, arranged to store meaningless data having a specified data length for use to provide the length of data in the body of dummy data, in said main storage means thereof; and data transfer is performed using a storage address of said dummy data storage area.

19. A data communication system according to any one of claims 1, 2 or 12 to 18 and wherein portions of said main storage means installed in said plurality of processing units are assigned as a global memory; and all of said processing units are arranged to access the global memory consisting of said main storage means to carry out parallel processing.

20. A data communication system according to any one of claims 1, 2 or 12 to 19 wherein each of said plurality of processing units comprises a main storage, a main control unit, a scalar unit including a central processing unit for executing scalar computation and a cache memory, a data transmission unit for transferring data to or from said data communication network means, a synchronous processing unit for extending synchronous control via a barrier network means, and an input/output unit for inputting or outputting data to or from external equipment.

21. A data communication system according to claim 20, wherein each of said plurality of

processing units further includes a vector unit for executing vector computation.

22. A data communication system according to any one of claims 1, 2 or 12 to 21 wherein said data communication network means is a crossbar network including buffer memories.

FIG. 1

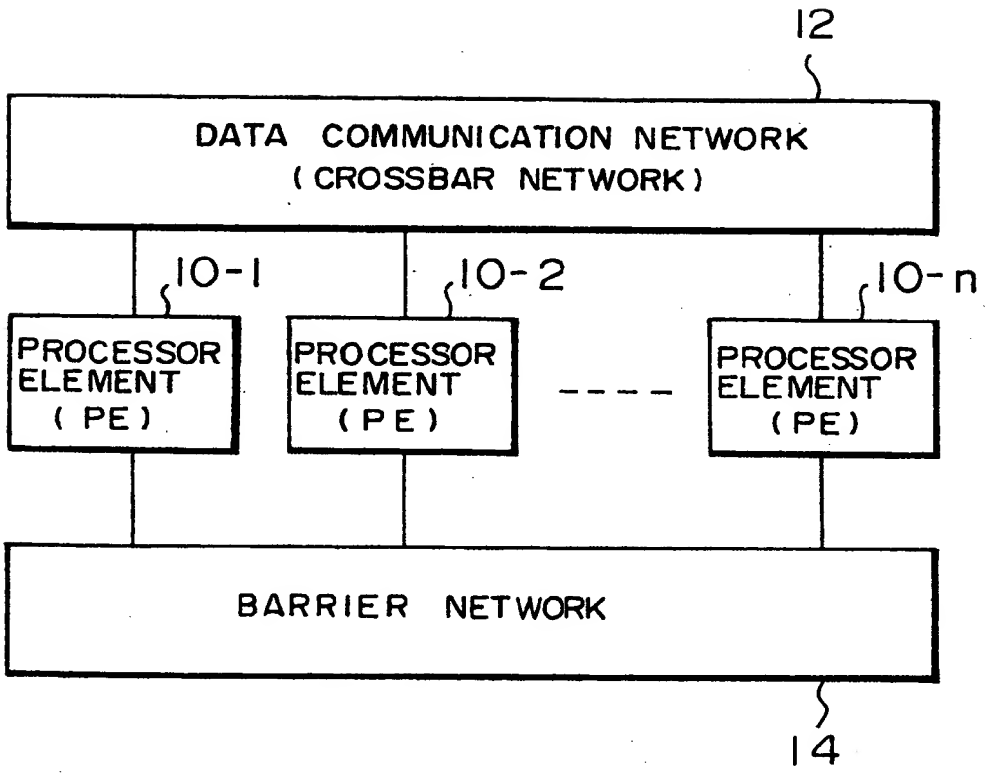

12
DATA COMMUNICATION NETWORK
(CROSSBAR NETWORK)
10-1
10-2
10-n
PROCESSOR ELEMENT (PE)
PROCESSOR ELEMENT (PE)
PROCESSOR ELEMENT (PE)
BARRIER NETWORK
14

FIG. 2

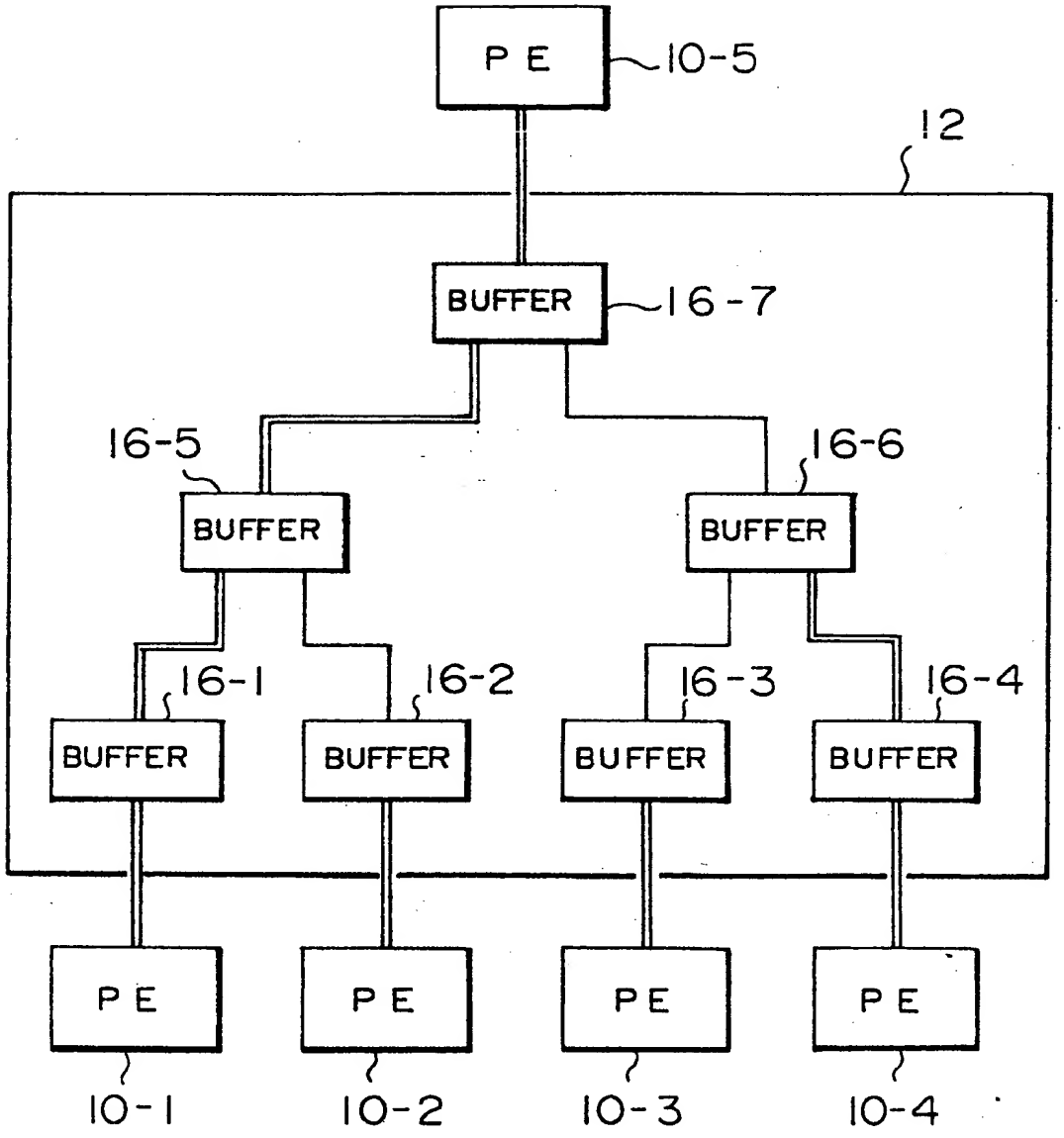

P E
10-5
12
BUFFER
16-7
16-5
16-6
BUFFER
BUFFER
16-1
16-2
16-3
16-4
BUFFER
BUFFER
BUFFER
BUFFER
P E
P E
P E
P E
10-1
10-2
10-3
10-4

FIG. 3

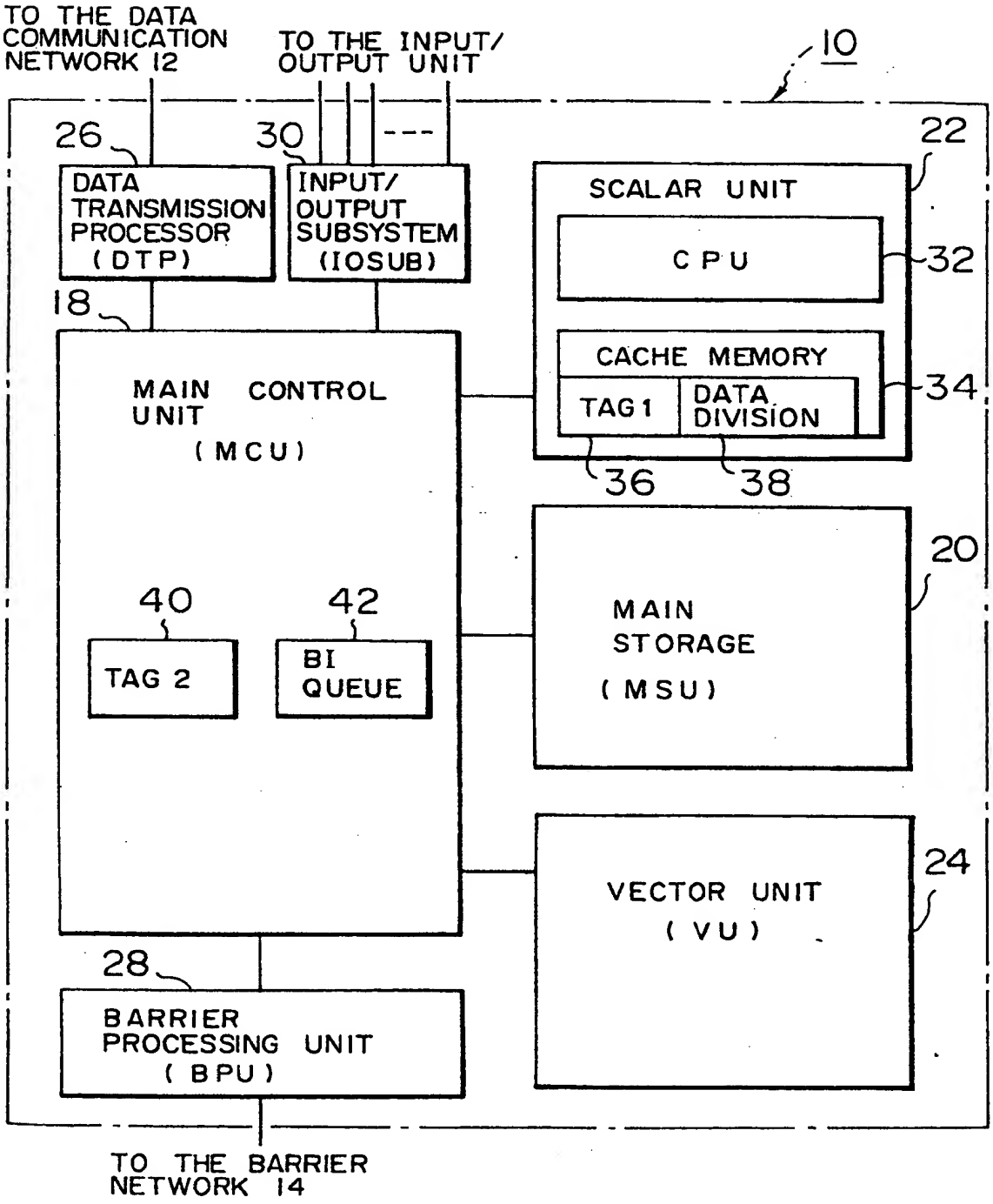

TO THE DATA COMMUNICATION NETWORK 12
TO THE INPUT/ OUTPUT UNIT
10
26
30
22
DATA TRANSMISSION PROCESSOR (DTP)
INPUT/ OUTPUT SUBSYSTEM (IOSUB)
SCALAR UNIT
CPU
32
18
CACHE MEMORY
TAG 1
DATA DIVISION
34
MAIN CONTROL UNIT (MCU)
36
38
20
40
42
MAIN STORAGE (MSU)
TAG 2
BI QUEUE
VECTOR UNIT (VU)
24
28
BARRIER PROCESSING UNIT (BPU)
TO THE BARRIER NETWORK 14

FIG. 4

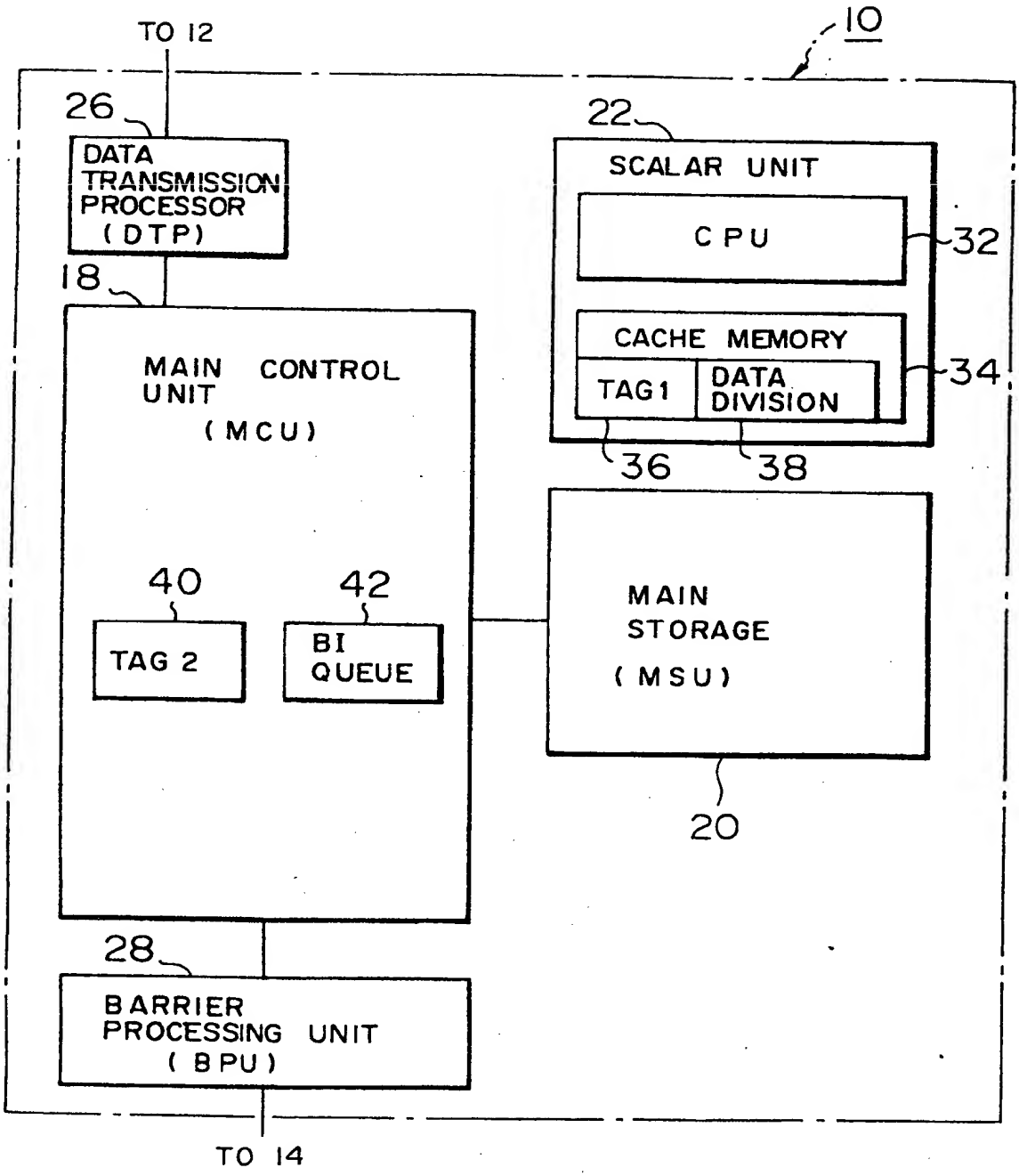

TO 12
10
26
DATA TRANSMISSION PROCESSOR (DTP)
22
SCALAR UNIT
CPU
32
18
CACHE MEMORY
TAG1
DATA DIVISION
34
MAIN CONTROL UNIT (MCU)
36
38
40
42
TAG 2
BI QUEUE
MAIN STORAGE (MSU)
20
28
BARRIER PROCESSING UNIT (BPU)
TO 14

FIG. 5

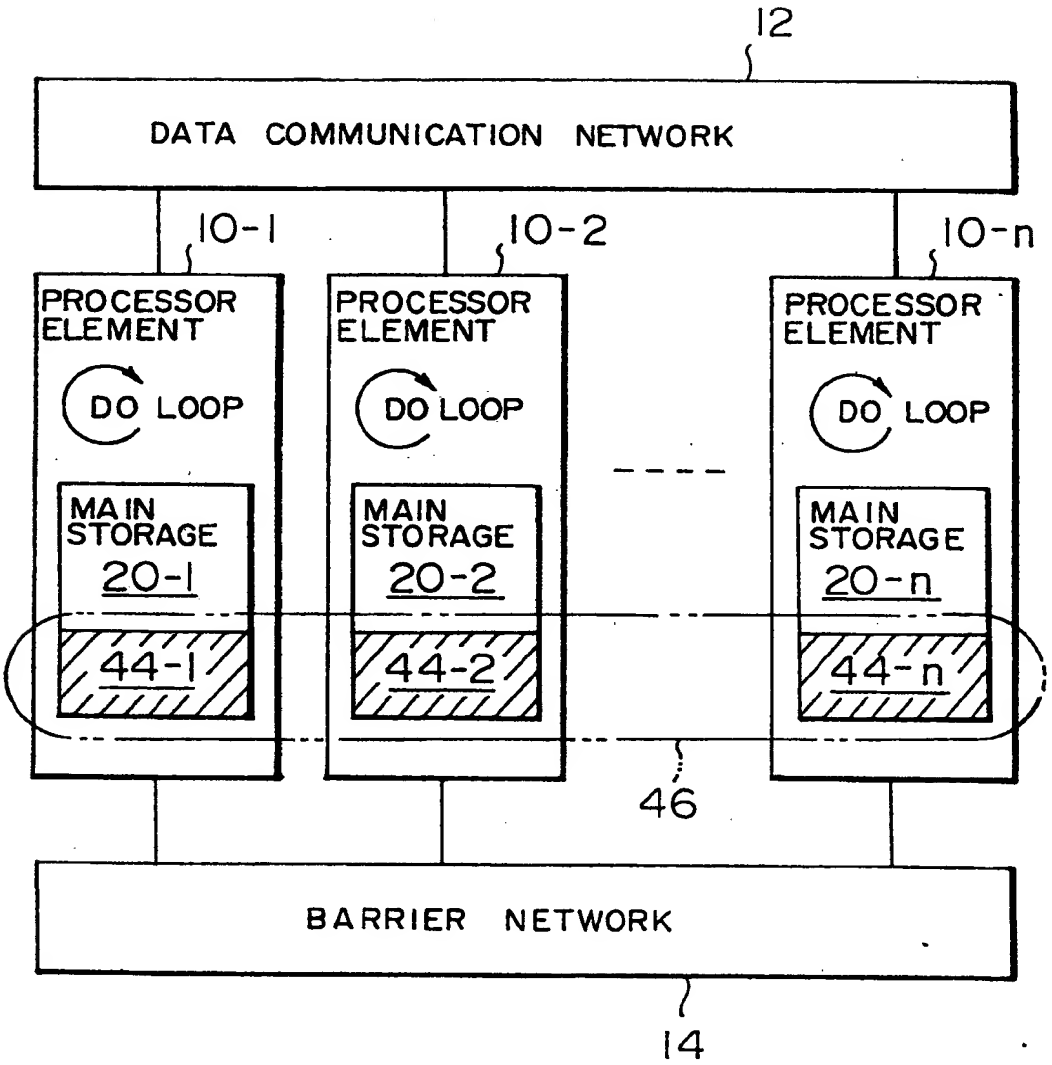

12
DATA COMMUNICATION NETWORK
10-1
10-2
10-n
PROCESSOR ELEMENT
DO LOOP
MAIN STORAGE
20-1
20-2
20-n
44-1
44-2
44-n
46
BARRIER NETWORK
14

FIG. 6

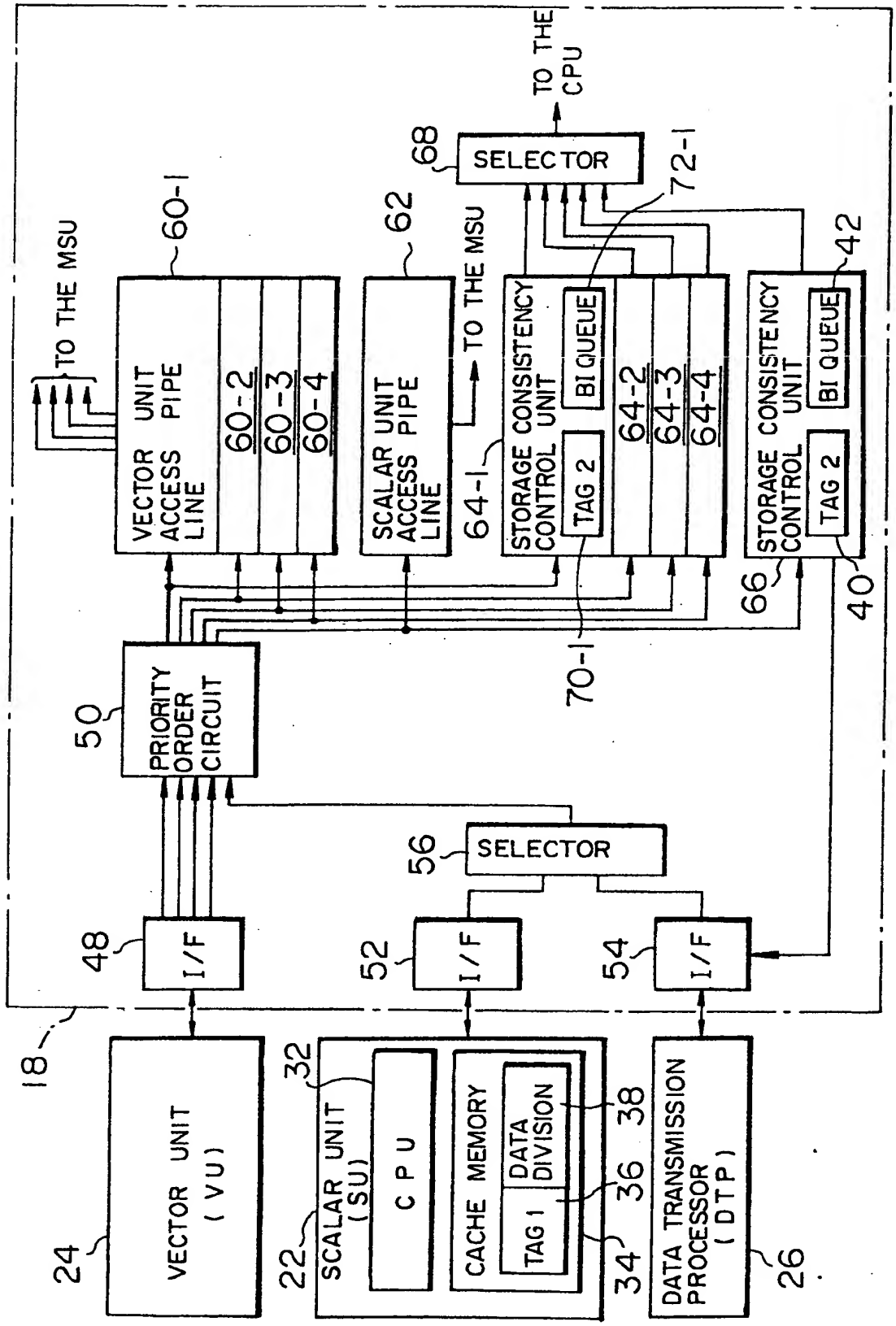

TO THE CPU
68
SELECTOR
72-1
60-1
62
TO THE MSU
42
TO THE MSU
VECTOR UNIT ACCESS PIPE
60-2
60-3
60-4
SCALAR UNIT ACCESS PIPE
64-1
STORAGE CONSISTENCY CONTROL UNIT
BI QUEUE
TAG 2
64-2
64-3
64-4
STORAGE CONSISTENCY CONTROL UNIT
BI QUEUE
TAG 2
66
40
70-1
50
PRIORITY ORDER CIRCUIT
56
SELECTOR
48
I/F
52
I/F
54
I/F
18
24
VECTOR UNIT (VU)
32
SCALAR UNIT (SU)
C P U
CACHE MEMORY
DATA DIVISION
38
TAG 1
36
34
22
DATA TRANSMISSION PROCESSOR (DTP)
26

FIG. 7

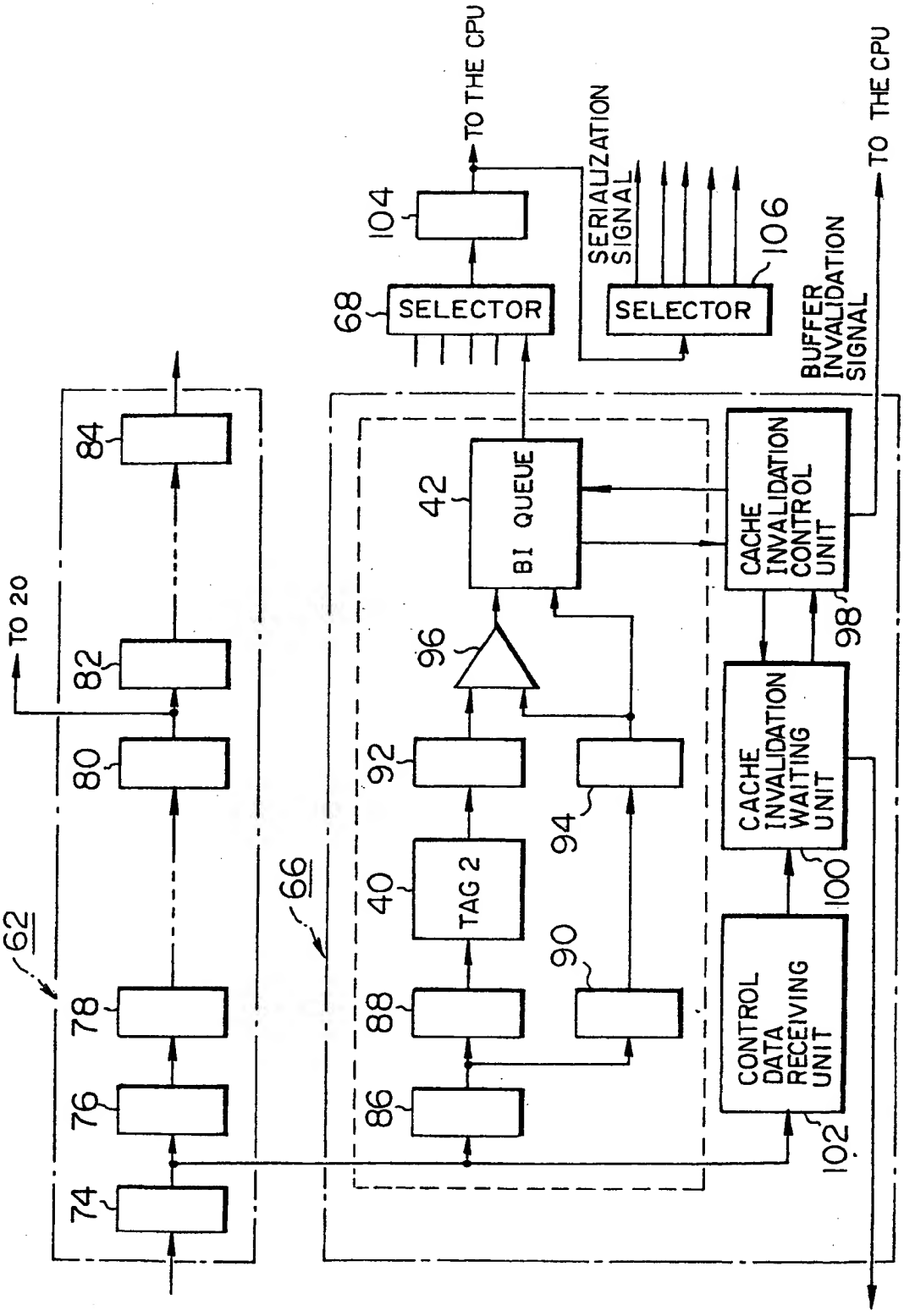

TO THE CPU
SERIALIZATION SIGNAL
BUFFER INVALIDATION SIGNAL
TO THE CPU
104
106
68
SELECTOR
SELECTOR
84
82
80
78
76
74
62
66
42
BI QUEUE
96
92
94
40
TAG 2
88
90
86
CACHE INVALIDATION CONTROL UNIT
98
CACHE INVALIDATION WAITING UNIT
100
CONTROL DATA RECEIVING UNIT
102
TO 20

FIG. 8

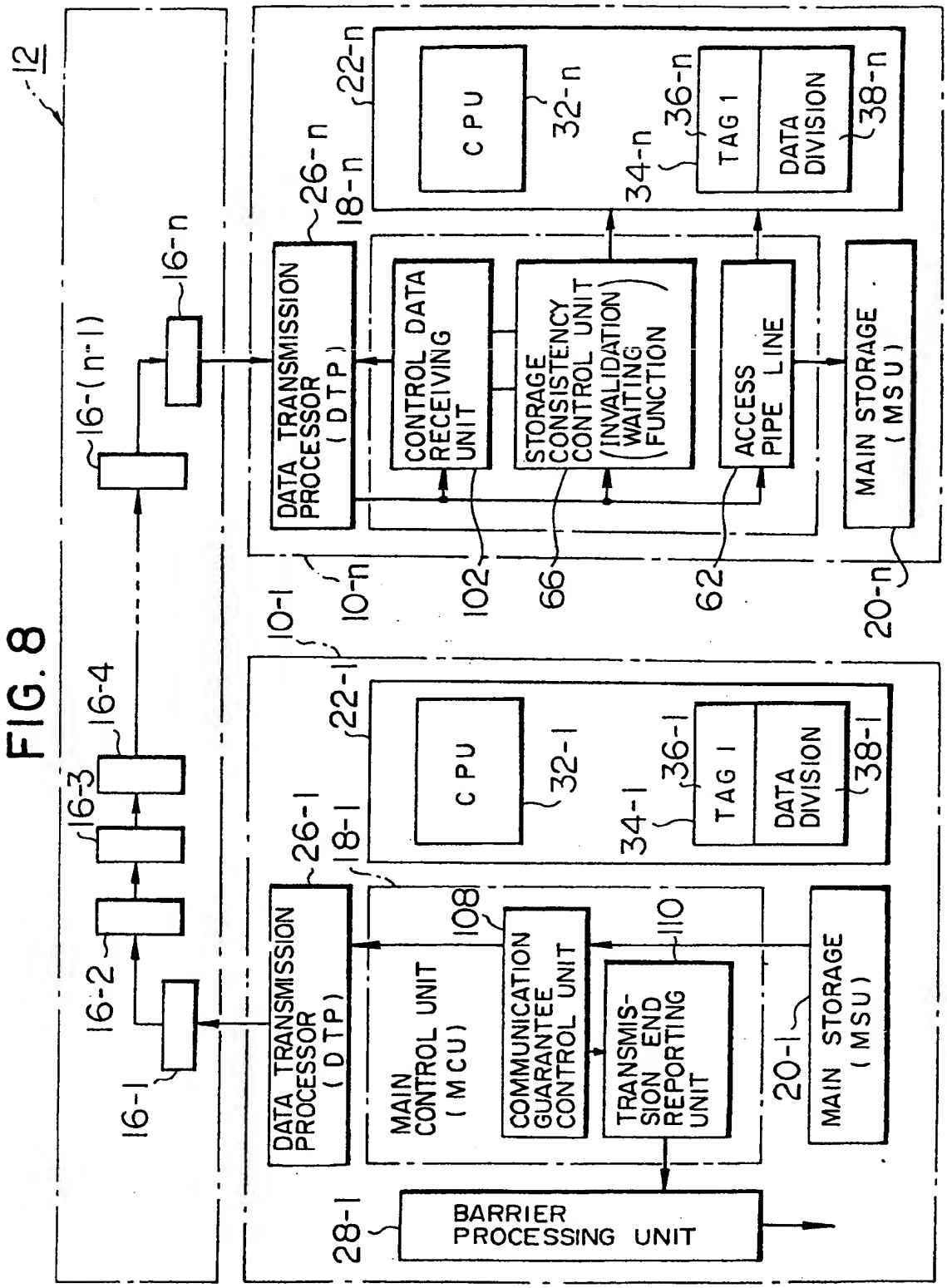

12
16-1
16-2
16-3
16-4
16-(n-1)
16-n
10-1
10-n
18-1
18-n
22-1
22-n
26-1
26-n
DATA TRANSMISSION PROCESSOR (DTP)
MAIN CONTROL UNIT (MCU)
CONTROL DATA RECEIVING UNIT
STORAGE CONSISTENCY CONTROL UNIT (INVALIDATION WAITING FUNCTION)
ACCESS PIPE LINE
MAIN STORAGE (MSU)
CPU
32-1
32-n
34-1
34-n
TAG 1
36-1
36-n
DATA DIVISION
38-1
38-n
102
66
62
20-n
108
COMMUNICATION GUARANTEE CONTROL UNIT
110
TRANSMISSION END REPORTING UNIT
20-1
28-1
BARRIER PROCESSING UNIT

FIG. 9

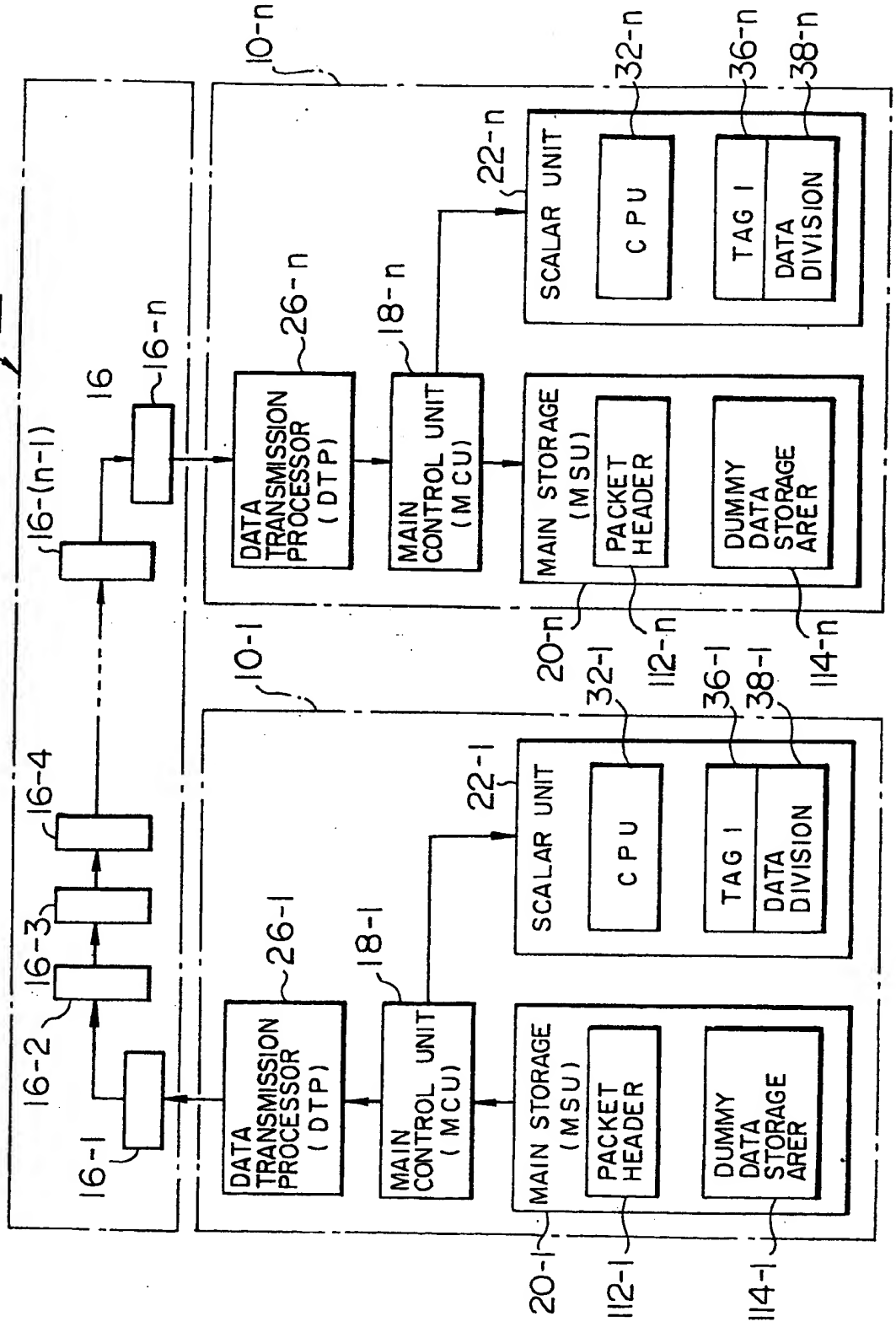

12
16
16-1
16-2
16-3
16-4
16-(n-1)
16-n
10-1
10-n
DATA TRANSMISSION PROCESSOR (DTP)
MAIN CONTROL UNIT (MCU)
MAIN STORAGE (MSU)
PACKET HEADER
DUMMY DATA STORAGE ARER
SCALAR UNIT
CPU
TAG
DATA DIVISION
18-1
18-n
20-1
20-n
22-1
22-n
26-1
26-n
32-1
32-n
36-1
36-n
38-1
38-n
112-1
112-n
114-1
114-n

FIG. 10

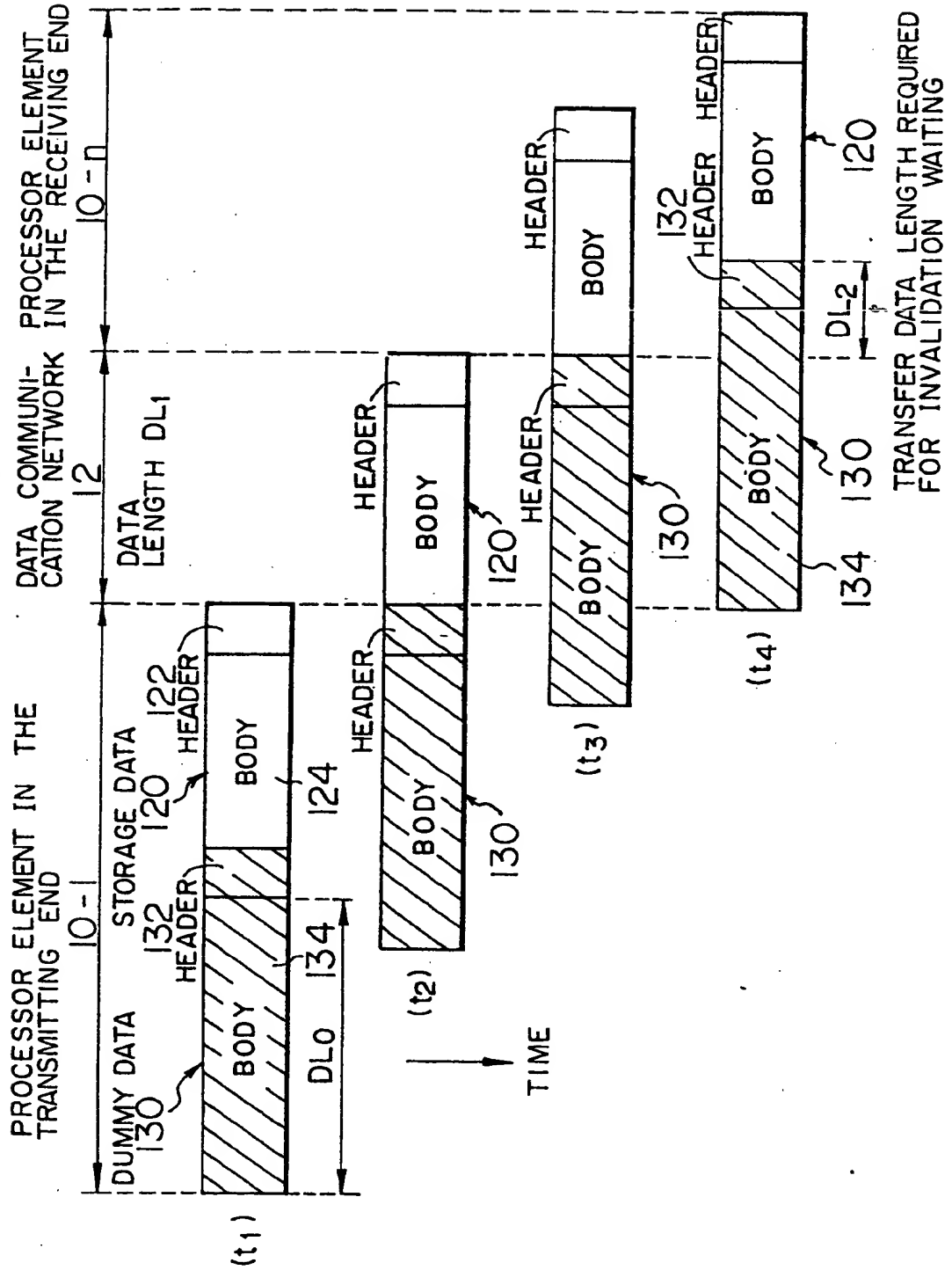
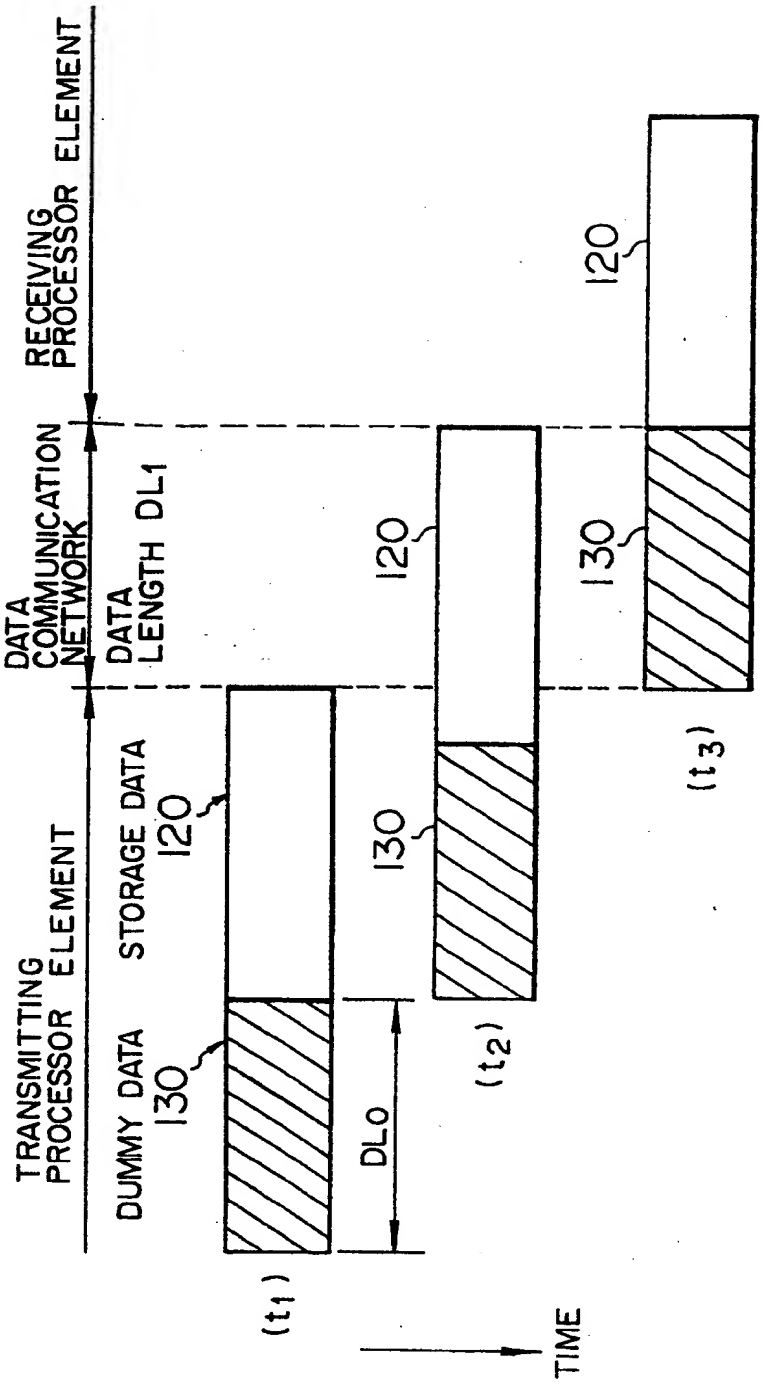

PROCESSOR ELEMENT IN THE TRANSMITTING END
10-1
DUMMY DATA
130
STORAGE DATA
120
132
HEADER
122
HEADER
BODY
124
DL0
134
BODY
(t1)
DATA COMMUNICATION NETWORK
12
DATA LENGTH DL1
HEADER
BODY
120
130
(t2)
TIME
HEADER
BODY
120
130
(t3)
PROCESSOR ELEMENT IN THE RECEIVING END
10-n
132
HEADER
HEADER
BODY
120
DL2
130
BODY
134
(t4)
TRANSFER DATA LENGTH REQUIRED FOR INVALIDATION WAITING

FIG. 11


TRANSMITTING PROCESSOR ELEMENT
DATA COMMUNICATION NETWORK
RECEIVING PROCESSOR ELEMENT
DUMMY DATA 130
STORAGE DATA 120
DATA LENGTH DL1
DL0
130
120
130
120
(t1)
(t2)
(t3)
TIME